Furcadia FileServer Communications

Saturday, March 06, 2010
6:23 PM

Download Procedure

1. Connection to **hades.furcadia.com** on port **6500**
2. Wait for greeting.
3. Greeting by the fileserver (2 being fileserver thread ID?): **10 Welcome to Furcadia file server [2], proceed to id.**
4. Login + file request: **RC Artex Password pp1234 0 5 1024**
   - a. Login Successful response: **44 2291 5 1024 0**
     - i. Receive 1024 bytes of data.
     - ii. Verify data integrity with CRC32 (unless you trust them not to).
       - 1) Data valid:
         - a) If last packet:
           - i) Extract the remainder from the last packet (it will have trailing zeroes to fit packet size)
         - b) Append data to the file.
         - c) Send confirmation to server: **PKOK 0**
         - d) If last packet:
           - i) Send termination command: **BYE**
           - ii) [optional] Await termination confirmation: **99 Log out sequence initiated.**
           - iii) Disconnect.
       - 2) Data invalid:
         - a) Send transfer termination command: **+MIS 0 12345678**
         - b) Receive transfer termination confirmation from fileserver: **46 File error**
         - c) Receive transfer continuation request: **47 12345678 Continue transfer**
         - d) Send transfer resume request: **SNCRC 0**
         - e) Receive transfer resume confirmation: **48 12345678 Cont**
       - 3) **GO TO 4.a.i**
   - b. Login Failed response: **31 Not logged in.**
     - i. **GO TO 4 (or Disconnect)**
   - c. File not found response: **41 File not found.**
     - i. **GO TO 4 (or Disconnect)**
5. Get disconnected for inactivity: **98 Disconnected for inactivity.**

Upload Procedure

1. Connection to **hades.furcadia.com** on port **6500**
2. Wait for greeting.
3. Greeting by the fileserver (2 being fileserver thread ID?): **10 Welcome to Furcadia file server [2], proceed to id.**
4. Send in the username: **USER Artex**
5. Wait for confirmation: **20 Username received, continue to pwd.**
6. Send in the password: **PASS Password**
   - a. Login Successful response: **30 User logged in, proceed.**
     - i. Set resume position: **SETR 0**
     - ii. Wait for confirmation: **90 UFP set.**
     - iii. Send upload request for a package: **SEND td303628270 519067**
     - iv. Wait for clearance: **50 0 Ready to receive the file.**
     - v. Send transmission details: **SNCSEND 303628270 0**
     - vi. Send a 1024-byte data packet (or several).
     - vii. Wait for fileserver confirmation of receiption.
       - 1) Reception confirmed: **55 0**
         - a) **GO TO 6.a.vi**
       - 2) Reception error (invalid data): **56 0**
         - a) *Restart upload from the failed packet*… [How exactly? From the beginning?]
       - 3) Transfer complete (last packet processed): **59 File transfer complete.**
         - a) Upload another file (**SETR 0**, …); or
         - b) Log out: **BYE**
         - c) Disconnect.
   - b. Login Failed response: **31 Not logged in.**
     - i. **GO TO 4, 6, or Disconnect.**
7. Get disconnected for inactivity: **98 Disconnected for inactivity.**

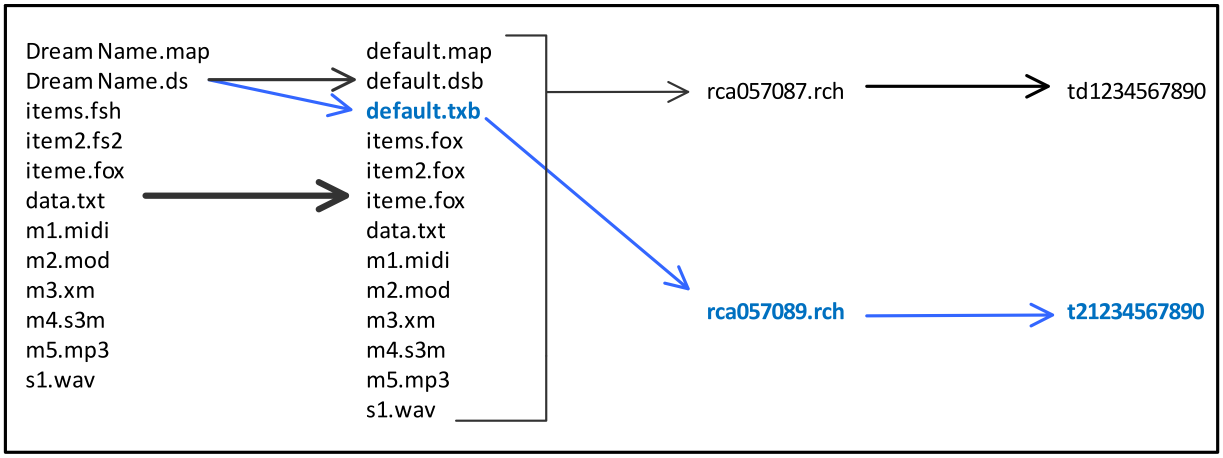
Data Packet Structure

Each data packet in the transfer is formed the following way:

Offset	Type	Size (bytes)	Description
0	string	2	"SC" - packet header identifier.
2	uint	4	Packet number (Little-Endian: 01 00 00 00 == 1)
6	uint	4	CRC32 summary of the data (Little-Endian?)
10	byte	PKTSIZE - 10	Contents of the packet

NOTE: The header data ("SC", the packet number, the CRC sum) are a part of the data packet! If the packet size is set to 1024 bytes, then 10 bytes out of it will contain the header and 1014 out of it will contain the actual data!

Dream Packaging & Upload Procedure



1. Filtering, Transformation, Optimization and Encryption$^{\text{(OPTIONAL)}}$:
    a. Any file that doesn't have a valid (for the dream package) extension will not be transferred. Others are processed and added to the RCH archives.
    b. The MAP and DS files now have the filename **default** instead of what they originally have.
    c. The DS file is split in two:
        i. **default.dsb** - DragonSpeak Binary file that contains all numeric data (line numbers, coordinates, variable addresses/references, string indexes, etc).
        ii. **default.txb** - DragonSpeak text list that contains all the text strings used to be in the DS file and a number through which they are referenced by the DSB file.
    d. All FSH and FS2 patch files are transformed to use the FOX format and exclude default graphics in default places. Each shape is set to replace a specific shape in the file's default counterpart.
    e. The MAP, DSB and FOX files become encrypted. Others are left as raw data in their original form.
2. Packaging:
    a. All files except for **default.txb** are stored in an RCH file in **%ALLUSERSPROFILE%\Dragon's Eye Productions\Furcadia\Temporary Files\**. The file has an **rca** prefix, a number and an **rch** file extension: **rca047087.rch**
    The contents of default.map (and possibly the others too) are XOR by 255 (0xFF): **data[i] = data[i] ^ 255**
    b. The **default.txb** file is stored in a similar, yet separate RCH file while its contents are also XOR by 255 (0xFF). The number part of the RCH file is incremented by 2: **rca047089.rch**
3. Upload:
    a. Done in a single session:
        i. **rca057087.rch -** (the dream package itself) is uploaded first, while its CRC32 sum is used as the dream's identity. The file is uploaded as td<crcsum> (i.e.: **td1234567890**).
        ii. **rca047089.rch** - uploaded next without closing the fileserver session. The prefix of this package is **t2** instead of **td** and the CRC32 sum right after it is the same as the dream's. This file will not be downloadable and will only be used by the server.
    b. If the upload was successful, a dream portal can then be registered with the **`tdgate <crcsum>** command on Furcadia.

Client-to-Server & Server-to-Client Command List

Client-to-Server Commands

RC <username> <passwd> <pkg name> <resume pos> <packets/burst> <packetsize>	Download	Log in and request a package (Request Content).
PKOK <packet #>	Download	Confirm successful packet reception.
SNCRC <packet #>	Download	Request transfer resume from specific packet.
+MIS <packet #> <packet crc32sum>	Download	Request transfer stop due to unsuccessful packet reception.
USER <username>	Upload	Log in with specific username.
PASS <password>	Upload	Log in with specific password.
SEND <package name> <package filesize>	Upload	Request upload clearance.
SETR #	Upload	Set resume position (units?).
SNCSEND <package crc32sum> <start packet?>	Upload	Begin upload notification.
BYE	Download/ Upload	Log out of the server.

Server-to-Client Commands

10	10 Welcome to Furcadia file server [0], proceed to id.	Greeting and identification as a Furcadia file server. (0 being thread ID?)
20	20 Username received, continued to pwd.	Username reception notification.
30	30 User logged in, proceed.	Login success confirmation (for Upload).
31	31 Not logged in.	Login failure notification (username/password wrong?).
41	41 File not found.	"File not found" error.
44	44 <file size> <packets per burst> <packet size> <start packet>	Download start notification.
46	46 File error	Download termination confirmation (sent several times after a +MIS command).
47	47 <packet crc32sum> Continue transfer	Server is ready to continue the transfer (SNCSEND command).
50	50 <start packet> Ready to receive the file.	SEND command confirmation - ready to receive the package from client.
55	55 <packet #>	Confirm successful packet reception (like PKOK).
56	56 <packet #>	Packet validation failed (like +MIS).
59	59 File transfer complete.	Notification that the file transfer is complete.
90	90 UFP set.	Upload File Position set - confirmation for SETR command.
98	98 Disconnected for inactivity.	Idle timeout - user is disconnected after this one.
99	99 Log out sequence initiated.	Confirmation for BYE command - client is to disconnect.

ToDo

- More factual info about resuming a download/upload.
- Samples or hexdump of a real download/upload procedure.
- Complementary file downloader/uploader scripts? :P